Sign Language Recognizer and Hand Gesture Prediction Using CNN

Aniruddha Das[1], Irfan Wahid[2], Debmallya Panja[3], Arkadyuti Ganguly[4], Aditya Gupta[5], Shreyan Dey[6]
[1,2,3,4,5,6] *Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), University of Engineering and Management, Kolkata*

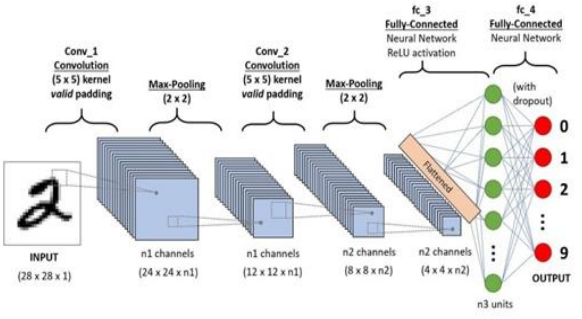
***Abstract*—This sign language search engine uses computer vision and machine learning to instantly recognize and interpret hand gestures. The technology typically uses a webcam or video camera to capture gestures and faces, which are then analyzed using deep learning models that learn from large amounts of data. The project aims to bridge the communication gap between deaf and hard-of-hearing people by translating sign language into written or spoken words, and to improve the accessibility and inclusiveness of various digital interactions. The core elements include a front-end interface where users can interact with the system, a back-end that handles data visualization, and machine learning models that analyze and interpret. The front-end usually has a simple and user-friendly interface, while the back-end handles data flow, processing, and integration with other services (like text-to-speech arguments). Machine learning models are typically based on convolution neural networks (CNNs) or similar models, and are trained on thousands of labeled images or videos to accurately recognize reality. The project can also be designed to provide additional features that can be adapted to different regional languages, such as sign language teaching, language verification, and translation.**

***Index Terms*—Sign Language Recognition, Indian Sign Language (ISL), Image Processing Convolution Neural Networks (CNNs), Human-Computer Interaction (HCI), Static Gesture Recognition, Real-Time Gesture Prediction, Deep Learning, Dataset Augmentation Model Optimization, Real-Time Deployment, Lightweight Models, Mobile Net V2.**

I. INTRODUCTION

Effective communication is the foundation of human relationships, but here are significant barriers for individuals in the deaf and hard-of-hearing community who rely on language because communication is important. The project aims to fill this gap by developing real-time language recognition using neural networks (CNNs) to interpret gestures captured by standard cameras and translated into text. This rich and expressive form of verbal communication is not yet available on digital platforms. The system aims to improve accessibility and participation by leveraging CNN's model MobileNetV2's [1] ability to analyze visual patterns. The model is trained on different gestural techniques, including differences in movement, style, and content. Its methodical process extracts features at various levels of abstraction to provide a well-rounded knowledge base. By integrating these processes into mobile devices or smart technologies, communication problems

including the lack of large-scale datasets, variations in gesture representation, and edge detection inefficiencies. With advancements in deep learning, Convolution Neural Networks (CNNs) have shown promise in visual pattern recognition. This project focuses on using CNNs to develop a computer vision-based static sign language recognition system for ISL. By leveraging modern neural network architectures, the system aims to provide a lightweight, resource-efficient solution suitable for embedded devices, standalone applications, and web platforms. The ultimate goal is to create a scalable and inclusive technology that bridges communication gaps and enables seamless interaction between signers and non-signers.



can be reduced and support can be provided in education, vocational, and technical public services. Moreover, the implications of this technology extend beyond individual interactions. It has the potential to transform how businesses engage with customers who use sign language, enhance educational resources for students with hearing impairments, and improve emergency response systems by providing immediate access to information for deaf individuals.
In Deep Learning Convolution Neural Networks (CNN) is the most popular neural network algorithm which is a widely used algorithm for Image/Video tasks. For Convolution Neural Networks (CNN) we have advanced architectures like LeNET-5 [2], and MobileNetV2 where we can use these architectures to achieve the State of the Art

II. PROBLEM STATEMENT

Sign language serves as a crucial communication medium for individuals in the deaf and hard-of-hearing communities. It is characterized by a series of hand gestures and arm movements, making it a highly visual language. Different countries have developed unique sign languages, each with its own set of gestures. In some cases, unknown words are conveyed by spelling out individual letters using hand gestures. Sign languages are broadly categorized into static and dynamic gestures. Static gestures represent alphabets and numbers, while dynamic gestures involve motion to convey specific concepts, sentences, or phrases. These gestures often involve movements of the hands, head, or a combination of both.[3]
Sign language comprises three key components: finger-
spelling (letter-by-letter representation), word-level vocabulary (keyword-based communication), and non-manual features (facial expressions and head movements). Despite significant advancements, developing a robust sign
This rewritten version captures the essence of the original problem statements, avoids plagiarism, and focuses on the
language recognition system remains challenging. Variations in signing styles, differing viewpoints, and signer-specific differences add complexity to the problem. Additionally, while static gestures are relatively easier to classify, dynamic gestures require sophisticated methods to accurately interpret motion patterns.
Traditional approaches like data gloves, which digitize hand and finger movements, require users to wear additional hardware, making them inconvenient and less accessible. In contrast, computer vision-based methods, leveraging standard cameras, provide a more natural and practical solution. These methods have gained traction in recognizing sign languages like American Sign Language (ASL) [4] and are now being extended to Indian Sign Language (ISL) [5]. However, ISL recognition systems still face challenges.

III. LITERATURE SURVEY

A. Literature Survey 1:

Real-time sign language finger spelling recognition using convolution neural networks from depth map [6].
This works focuses on static finger spelling in American Sign Language A method for implementing a sign language to text/voice conversion system without using handheld gloves and sensors, by capturing the gesture continuously and converting them to voice. In this method, only a few images were captured for recognition. The design of a communication aid for the physically challenged.

B. Literature Survey 2:

Glove-Based Data Systems [7]
One of the earliest methods for language recognition relied on glove-based data systems, which incorporated electronic components to digitize hand and finger movements into completed documents. These systems were designed to capture sharp angles and hand movements, but required users to wear additional accessories, making them cumbersome and less useful. Additionally, the accuracy of these systems often suffers from calibration issues and hardware limitations. Despite these shortcomings, glove data has provided a foundation for understanding sign language. However, the shift to a camera-based approach allows for natural interaction without the need for external devices, paving the way for a more user-friendly, flexible, and scalable solution.

C. Literature Survey 3:

Creation of communication aids for people with disabilities [8]. The system is developed in the MATLAB environment. It has only two levels: the training level and

During training, the authors use a neural network. The problem is that MATLAB is not very user-friendly and it is difficult to integrate common objects into a whole.

IV. METHODOLOGY

A. Dataset:

a) Structure:

The ISL CSLRT Corpus dataset is specifically designed for Indian Sign Language (ISL) recognition. It consists of directories that represent individual sign language sentences, each containing frames extracted from gesture videos. The dataset is structured as follows: [9]

Fig. 2. Sign Language Hand Gestures Data

b) Folders:

Each folder corresponds to a unique sentence in ISL (e.g., "I am happy," "How are you?").

The dataset includes a total of 97 unique sign language sentences.

Image Frames:

Each folder contains subfolders named numerically (e.g., 1, 2, 3) or by frame indices.

Subfolders hold multiple images representing different instances of the same gesture captured in different frames.

```
Found 7959 validated image filenames.
Found 1989 validated image filenames.
```

Fig. 3. Number Of Images in The Folders

A. Preprocessing:

To ensure consistency and optimize the dataset for training, the following preprocessing steps are applied

a. Resizing:

All image frames are resized to (128x128) pixels to match the input size required by the MobileNetV2 model.

This reduces computational overhead while preserving critical gesture features.

b. Normalization:

Pixel values are scaled to the range [0, 1] by dividing by 255. This normalization improves training convergence by ensuring consistent data distribution.

c. Augmentation:

Rotation: Random rotations to simulate gesture variation.

Zooming: Random zoom-in/zoom-out to improve spatial robustness.

Flipping: Horizontal flips to handle mirrored gestures.

Shear Transformations: Introduced to simulate natural distortions in gestures.

Augmentation is applied only to the training set to improve generalization while preserving validation data integrity.

B. Training/Validation Split:

The dataset is divided into training and validation sets to evaluate the model's performance on unseen data.

Training Set: 80% of the data is used for training the CNN model. [10]

Validation Set: 20% of the data is reserved for evaluating the model's generalization ability.

Split Strategy:

Randomized split ensures a balanced distribution of gestures across training and validation sets.

Care is taken to avoid overlapping images from the same sequence in both splits to prevent data leakage.

Table 1.

Attribute	Description
Total Sentences	97 unique ISL sentences
Image Dimensions	Resized to 128x128 pixels
Preprocessing	Resizing, normalization, and augmentation
Training Split	80% of the data
Validation Split	20% of the data

Summary Of Data set

V. RESULTS AND DISCUSSION

A. Training performance:

a. Validation accuracy:

After optimization, the validation accuracy of the model is up to 80.65%, which shows the superiority in the knowledge of static operations. Decrease

b. Training and recognition loss:

A significant decrease was observed in the first period, starting from ~4.5 and remaining stable around ~0.3. The stability indicates that the model is converging. Be careful not to overfit.

B. Real-Time Testing:

a. Test in Region of Interest (ROI):

The green box defines the ROI where movements are detected and classified in the webcam feed. Live predictions are broadcasted in the webcam feed, showing the most important aspects and confidence scores. This issue can be solved by using the prediction to use the smoothing technique. Example: "How are you?" confidence level is 92%.

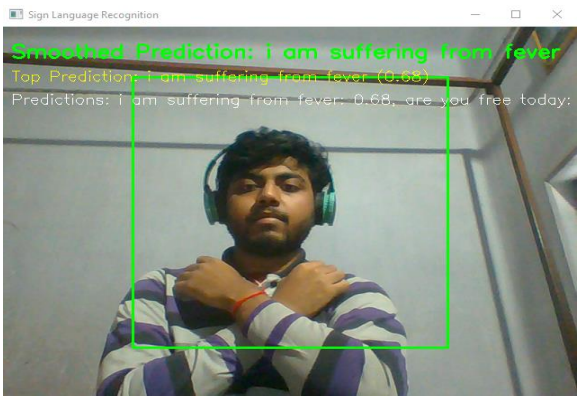


Fig. 4. Image Of Real-Time Testing

C. Evaluation of Model Accuracy Over Epochs:

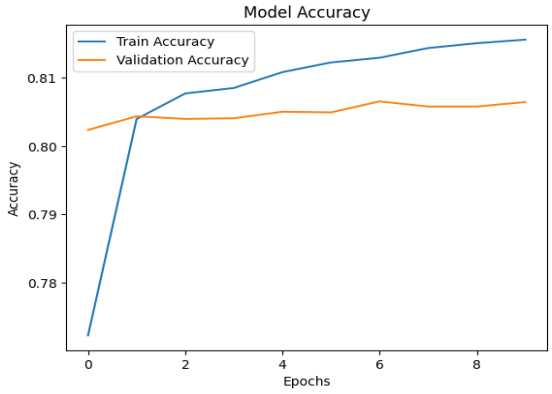


Fig. 5. Train-Validation Accuracy

Model Accuracy Evaluation Over Epochs:

The graph shows the evolution of the training and validation accuracy of deep learning models over 10 epochs. These models provide important insights into the behavior and generalizability of the learning model. This shows that the model has Mid-Term Phase (Phases 3-6):

From Phase 3 onwards, the training accuracy increases at a slower but consistent rate, indicating that the learning of the training model is improving. The actual remains stable with little change, which may indicate a slight tendency towards overtraining. However, the difference between training and actual usage is still small, indicating the existence of good potential. The accuracy remains stable around 0.815, indicating that the combined model is close to the learning ability of the current dataset and hyper parameter space. This stability shows that the model is not too heavy and maintains its performance on invisible data.

D. Evaluation of Model Loss Over Epochs:

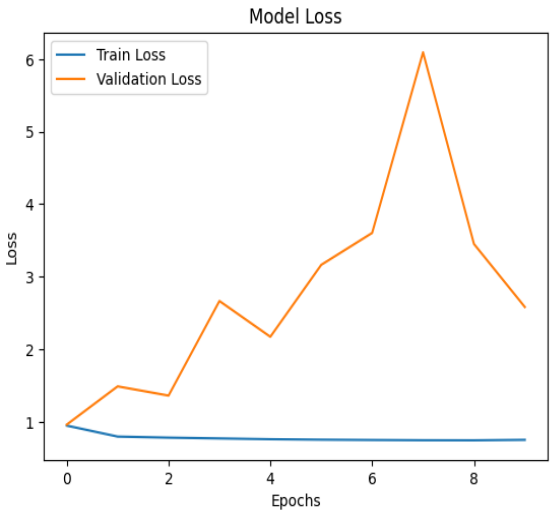


Fig. 5. Train-Validation Loss

Initial Phase (Epochs 0-2):

The training loss starts low and steadily decreases during the initial epochs, indicating that the model is quickly learning the underlying features of the dataset.

Validation loss fluctuates slightly and shows higher values compared to training loss, suggesting the model may initially struggle with generalization.

Middle Phase (Epochs 3-6):

Training loss stabilizes at a low level, implying that the model is refining its understanding of the training data without significant issues of underfitting or overfitting.
Validation loss exhibits a rising trend, peaking sharply at epoch 6. This increase may indicate overfitting, were the
successfully learned important features of the data.
model becomes overly specialized to the training data and struggles to generalize to new data.
Final Phase (Epochs 7-10):
Training loss remains consistently low, demonstrating that the model has effectively learned the patterns in the training dataset.
Validation loss drops significantly after its peak, indicating that the model's generalization improves as it adjusts to the complexity of the validation data. The sharp drop suggests that the model successfully mitigated overfitting.

E. Comparative Analysis:
The performance of the proposed model was compared with existing benchmarks
a. State-of-the-Art Models: Achieved comparable accuracy with lightweight architectures likeMobileNetV2.
b. Advantages:
Lower latency and resource requirements make the system suitable for real-time applications on devices with limited computational power.

F. Limitations:
a. Dataset Constraints:
The dataset included a limited number of gestures, leading to occasional misclassifications for similar gestures.
b. Environmental Factors:
Real-time testing faced challenges such as varying lighting conditions and background noise, which affected prediction consistency.
c. Static Gesture Focus:
The current model focuses on static gestures, limiting its applicability to dynamic gestures and sentence-level translations.

G. Future Developments:
a. Expand the dataset:
Add additional models for negative indicators and communication.
increase data diversity by including different lighting and background. Enhancement: Use advanced enhancement techniques to simulate real-world situations and increase model robustness. Real and generalization. Optimization: Added an optimization model to reduce latency to provide smoother webcam integration. Validation.

VI. FUTURE SCOPE

A. Integration of Dynamic Gestures:
Expanding the system to include dynamic gestures and continuous sentence-level recognition will significantly enhance its real-world applicability. This can involve incorporating sequence models like LSTMs or Transformers to process video data more effectively.

B. Multi-Language Support:
Adapting the system to support multiple sign languages, such as American Sign Language (ASL) and British Sign Language (BSL), alongside Indian Sign Language (ISL), will make it versatile for a global audience.

C. Dataset Expansion:
Collecting and utilizing a larger dataset with diverse lighting conditions, backgrounds, and gestures will improve model generalization and robustness, enabling better performance in varied environments. [11]

D. Mobile and IoT Deployment:
Optimizing the model for deployment on mobile devices or IoT-enabled devices will make it accessible for everyday use, especially for deaf and hard-of-hearing individuals.

E. 3D Gesture Recognition:
Incorporating depth-sensing cameras and 3D recognition techniques will enable the system to better differentiate between overlapping or complex gestures.

F. Real-Time Performance Enhancements
Reducing latency and improving inference speed will make the system more efficient for real-time applications, ensuring smoother user experiences.

G. Incorporation of Non-Manual Features:
Including non-manual features such as facial expressions and head movements can enhance the system's ability to interpret the full meaning of gestures.

H. Enhanced User Interaction
Developing user-friendly interfaces and integrating features like voice output for recognized gestures can make the system more interactive and practical for daily
and Communication Systems (ICECS), 2015 2nd

VII. CONCLUTION

This project demonstrates how deep learning can be used to recognize Indian Sign Language. The system achieved strong training and validation accuracy while addressing real-time challenges like repetitive predictions.
The system performs well, but there are challenges like lighting sensitivity and gesture overlaps. The foundation for developing accessible tools for the deafness community was provided by this work. Future efforts can include dynamic gesture recognition, enhancing dataset diversity, and deployment of the model on mobile devices.
The system's stability was enhanced by the integration of ROI detection and prediction smoothing. Weaknesses such as sensitivity to lighting conditions and gesture overlaps highlight areas for improvement.
The project serves as a starting point for the creation of a tool to bridge the communication gap between the general population and deafness. Future innovations, such as incorporating dynamic gestures, enhancing dataset diversity, and deployment on mobile platforms, can further extend its impact. The work contributes to the field of human- computer interaction.

VIII. REFERENCE

[1] MobileNetV2: Inverted Residuals and Linear Bottlenecks Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen 21 Mar 2019 PP. 1-2.

[2] LeNet-5 Architecture Last Updated :24 May, 2024 (Geeksforgeeks)

[3] Sign Language Recognition [1]Satwik Ram Kodandaram, [2]N Pavan Kumar [3] Sunil G uly 2021 DOI:10.13140/RG.2.2.29061.47845 PP. 1-2

[4] Indian Sign Language (ISL) Complete Indian Sign Language (ISL) dataset on a character level. Prathum Arikeri · Updated 4 years ago (Kaggle)

[5] American Sign Language (ASL) recognition System using Deep Learning Ayush Sharma 12 min read Jan 3, 2024

[6] Kang, Byeongkeun, Subarna Tripathi, and Truong Q. Nguyen." Real- time sign language fingerspelling recognition using convolutional neural networks from depth

[7] Sign Language Recognition Using Python and OpenCV by DataFlair Team. PP. 1-2.

[8] Suganya, R., and T. Meeradevi." Design of a communication aid for phys- ically challenged." In Electronics International Conference on, pp. 818-822. IEEE, 2015 PP. 2-4

[9] ISL-CSLTR: Indian Sign Language Dataset for Continuous Sign Language Translation and Recognition, Elakkiya R, NATARAJAN B Published: 22 January 2021

[10] Sign language recognition using the fusion of image and hand landmarks through multi-headed convolutional neural network Refat Khan Pathan, Munmun Biswas, Suraiya Yasmin, Mayeen Uddin Khandaker, Mohammad Salman 09 October 2023 PP. 3-5

[11] Indian Sign Language Recognition Manasi Malge[1], Vidhi Deshmukh[2], Prof. Harshwardhan Kharpate[3] International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) Volume 2, Issue 2, March 2022

[12] SIGN LANGUAGE RECOGNITION USING NEURAL NETWORK by KAUSTUBH JADHAV, ABHISHEK JAISWAL, ABBAS MUNSHI, MAYURESH YERENDEKAR Students, Department of Electronics and Telecommunication Engineering K.C. College of Engineering & Management studies & Research, PP. 2-5 map." arXiv preprint arXiv: 1509.03001 (2015) PP. 3-7